Supplementary Material for LabelFusion: A Pipeline for Generating Ground Truth Labels for Real RGBD Data of Cluttered Scenes

1 Comparison with Human Labeling of Single Frame

To approximately quantify the quality of the data generated by our pipeline, and the speed of labeling, we compared with a traditional technique of labeling one image with a polygon of the segmented object (Figure 1). We randomly chose two images from our dataset, and used [1] to label them by hand. A side-by-side comparison of the human labeling and the label generated from our pipeline are provided below. Human labeling using [1] took approximately 10 minutes per frame. With our method we spent approximately 60 seconds of human input to label each of these scenes, but this is amortized over 1,000 views of this scene. Accordingly the human time per label (Figure 1, bottom row) is approximately four orders of magnitude less for our method.

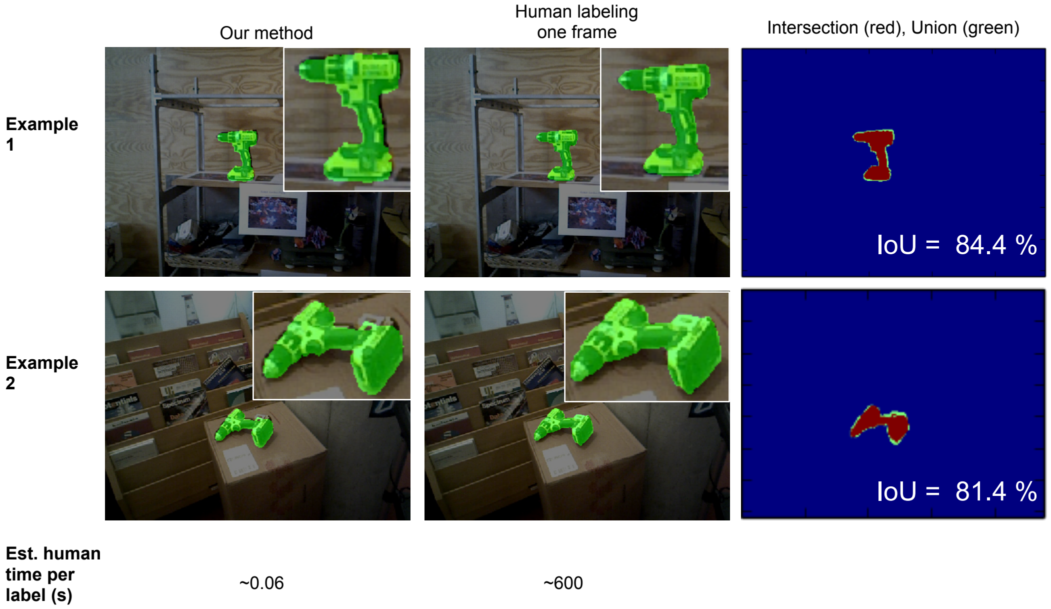

Figure 1: Comparisons of our method (left) vs. human single-frame labeling (middle), for two example images. The insets show a zoomed-in cropped view around the object. For the images, we respectively compute the intersection over union (IoU) for the drill mask at 84.4% and 81.4% (right). Our method has approximately a four orders of magnitude advantage in terms of human labeling time per frame (bottom row).

2 Experimental Details

Here we expand on experimental details that due to space constraints we could not fit into the main manuscript.

2.1 Object Set and Object Meshes

In total we used a set of 12 object meshes. We tested a variety of methods for acquiring object meshes. The highest quality meshes we produced (oil bottle, phone, red robot, and drill) were from our handheld Artec 3D scanner. We also used a tabletop, spinning 3D scanner (toothpaste) which was more difficult to use. For the tissue box, we simply measured it by hand and created a box primitive mesh. Other meshes were obtained from others' datasets, including the blue funnel from [2] and the cracker box, tomato soup, spam, and mug from the YCB object set [3].

2.2 Segmentation Network Training

Used a TensorFlow reimplementation [4] of DeepLab [5], but without the CRF post-processing step. We implemented CRF post-processing but found this to not improve results, due to the challenging occlusions and neighboring objects with similar color textures. We began training our models with the weights provided by [4], which were pre-trained on the PASCAL VOC dataset. All images from the native Asus Xtion resolution, 640 × 480, were downsized to 480 × 360 for training. Parameters used for training were the defaults in [4] for full-network training: 2.5 e-4 step size, 0.9 momentum, 20,000 steps, 2 batch size. The one exception was our "how many views?" experiment, for which 100,000 steps at a batch size of 2 was used in order to allow the potential benefit of the larger datasets. All models were trained on a GTX 1080; training completed in approximately 2.5 hours for 20,000 steps, and 12.5 hours for 100,000 steps.

2.3 Training Data Details

The empirical evaluations were performed with variations on two primary groups of training data. These two groups of training data did not encompass all of our data we generated (for example, they did not include all of the objects we have generated data for, or all the environments), but they did comprise a focused subset which allowed focused comparisons.

More visuals of these scenes are provided in our video (see website at `labelfusion.csail.mit.edu`).

2.3.1 Multi-object Training Set

A set of 51 total scenes were used for the multi-object training experiments. All of these scenes were taken the same day, over the course of a few hours, and were each taken with the same pre-programmed motion of the Kuka IIWA arm with mounted Asus Xtion camera. Lighting was kept constant throughout all experiments. From each scene were taken 135 +/- 1 seconds of data at 30 Hz, giving 4,000 frames per scene. The six objects for these experiments were: oil bottle, drill, tissue box, spam, cracker box, and the blue funnel.

In summary, the total number of training and test scenes available were:

- 18 single-object training scenes (3 scenes each for each of 6 objects)
- 18 multi-object training scenes (each with all 6 objects)
- 6 single-object test scenes (1 scene each for each of 6 objects)
- 9 multi-object test scenes

2.3.2 Drill Training Set

A set of 61 total scenes were used for the experimentation with a wide variety of backgrounds. These scenes were mostly taken by handheld data collection, except for 3 that were from an KUKA-arm-mounted data collection. Each handheld scene comprised of 34 seconds of 30 Hz data for approximately 1,000 frames.

References

[1] P. Tangseng, Z. Wu, and K. Yamaguchi. Looking at outfit to parse clothing. Mar 2017. URL http://arxiv.org/abs/1703.01386v1.

[2] J. M. Wong, V. Kee, T. Le, S. Wagner, G.-L. Mariottini, A. Schneider, L. Hamilton, R. Chipalkatty, M. Hebert, D. M. S. Johnson, J. Wu, B. Zhou, and A. Torralba. SegICP: Integrated Deep Semantic Segmentation and Pose Estimation. *ArXiv e-prints*, Mar. 2017.

[3] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar. Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set. *IEEE Robotics & Automation Magazine*, 22(3):36–52, 2015.

[4] DrSleep. DeepLab-ResNet-TensorFlow, https://github.com/DrSleep/tensorflow-deeplab-resnet. https://github.com/DrSleep/tensorflow-deeplab-resnet.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016.